GMOS: Grounding Moving Object Segmentation in 3D Space and Time

Junyu Xie[1] **Tengda Han**[1] **Weidi Xie**[1,2] **Andrew Zisserman**[1]
[1]Visual Geometry Group, Department of Engineering Science, University of Oxford, UK
[2]SAI, Shanghai Jiao Tong University, China
{jyx,tengda,weidi,az}@robots.ox.ac.uk
https://www.robots.ox.ac.uk/vgg/research/gmos/



Figure 1: **Grounding Moving Object Segmentation (MOS) in 3D space and time. Left:** GMOS grounds MOS in 3D *space*, enabling reliable segmentation under challenging viewpoints and heavy depth parallax from moving cameras. **Right:** GMOS also grounds MOS in *time*: conventional MOS masks every object that moves at some point in the sequence, regardless of whether it is currently in motion. In contrast, our proposed *MOS-I* ("I" for Instantaneous) protocol requires predicting, for every frame, which objects are *actively moving* at that instant. This is illustrated in the bottom row, where a small bird lands, stops, then takes off, while the tree branch remains static throughout.

Abstract

Moving Object Segmentation (MOS) aims to discover, segment, and track objects that move independently of the camera. Current MOS methods, however, exhibit two fundamental limitations: they rely on pre-computed 2D auxiliary modalities such as optical flow or point trajectories that lack 3D geometric information, and they treat motion as a sequence-level attribute, overlooking the instantaneous motion state of each object. We address both by grounding MOS in 3D space and time, and propose GMOS, a framework that operates directly on RGB video to produce 3D-aware, temporally fine-grained segmentation of multiple moving objects, alongside a foreground–background variant GMOS-S for faster deployment. To support training and evaluation in this regime, we curate GMOS-2K, a dataset of 2,210 real-world videos with per-object temporal motion annotations drawn from five established Video Object Segmentation (VOS) benchmarks, and formalise MOS-I ("I" for instantaneous), a temporally fine-grained evaluation protocol with three complementary metrics. GMOS achieves state-of-the-art results across MOS, MOS-I, and Unsupervised VOS benchmarks, while running significantly faster than prior multi-object MOS methods and supporting online inference for streaming deployment.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Comparison with Moving Object Segmentation (MOS) baselines. *Auxiliary inputs typically involve optical flow, 2D tracks and depths. The runtime measurement setting is detailed in Appendix B.

Method	Multi-object	3D-aware	Temporally fine-grained	Online	No aux. inputs*	Per-frame runtime (s)
FlowSAM [76]	✓	✗	✗	✗	✗ (flow)	1.79
SegAnyMo [24]	✓	✓	✗	✗	✗ (depth, track)	4.06
GeoMotion [23]	✗	✓	✗	✗	✗ (flow)	0.87
GMOS-S	✗	✓	✓	✓	✓	0.26
GMOS	✓	✓	✓	✓	✓	0.38

1 Introduction

Moving object segmentation (MOS) aims to discover, segment, and track objects that move independently of the camera. This capability is foundational to applications such as autonomous driving [5, 38], camouflaged object detection [32, 79], and video surveillance [50, 73], and has become increasingly central to recent 3D/4D scene reconstruction methods [21, 40] that require motion masks to separate dynamic objects from the static scene.

Current MOS methods exhibit two fundamental limitations, as outlined in Tab. 1. *First*, they are not 3D-aware and remain fragile under large camera motion or depth parallax. This stems from their heavy reliance on pre-computed 2D auxiliary modalities (*e.g.* optical flow [41, 74, 79] or point trajectories [24, 29]), which lack 3D geometric information and incur substantial pre-processing overhead. *Second*, existing benchmarks assume that objects move throughout the video, leading methods to treat motion as a *sequence-level* attribute and to overlook the instantaneous motion state of each object. This coarse temporal granularity precludes online and real-time deployment.

We address both limitations by grounding MOS in *space* and *time*. Our approach rests on two observations: (*i*) the geometric representations learnt by recent feed-forward 3D reconstruction models already encode sufficient information to disentangle camera and object motion, and (*ii*) independent object motion is reliably identifiable from a short temporal window ($\sim$0.5 s).

Building on these insights, we propose GMOS, a proposer–propagator framework that operates directly on RGB video without any auxiliary pre-computed modalities. A per-frame *proposer* fuses geometric features from a π^3 encoder [72] with segmentation features from a SAM2 encoder [60], jointly producing object mask proposals and per-frame motion-state predictions. A SAM2-based *propagator* then links these proposals into consistent tracks across frames. We further introduce GMOS-S ("S" for *single*), a streaming foreground–background variant that predicts all moving objects as a *single binary* mask and dispenses with the propagator for even faster inference.

To support MOS in a more geometrically grounded and temporally fine-grained manner, we curate GMOS-2K, a collection of 2,210 real-world videos drawn from five established VOS benchmarks and annotated with per-frame motion states for every object. The subsets span complementary domains, including heavily occluded scenes, hand–object interactions, and wildlife and human-centric clips. Alongside this dataset, we formalise MOS-I ("I" for *instantaneous*), a temporally fine-grained MOS evaluation protocol. As illustrated in Fig. 1 (right), MOS-I scores segmentation masks only on frames where objects are actively moving, and penalises false-positive predictions on static and background objects.

In summary, our contributions are as follows:

- We propose GMOS, a proposer–propagator framework that delivers 3D-aware, temporally fine-grained MOS from RGB video alone and supports online inference. We also introduce GMOS-S, a foreground–background variant for even faster deployment.
- We introduce a confidence-based training objective that automatically down-weights noisy frame-level motion annotations near static–moving transitions.
- We curate GMOS-2K, a dataset of 2,210 real-world videos drawn from five established VOS benchmarks and annotated with per-object temporal motion labels, providing a unified resource for both training and evaluating temporally fine-grained MOS methods.
- We formalise MOS-I, a temporally fine-grained evaluation protocol for MOS, together with three complementary metrics that jointly measure segmentation accuracy on actively moving frames and robustness to false-positive predictions on static and background objects.

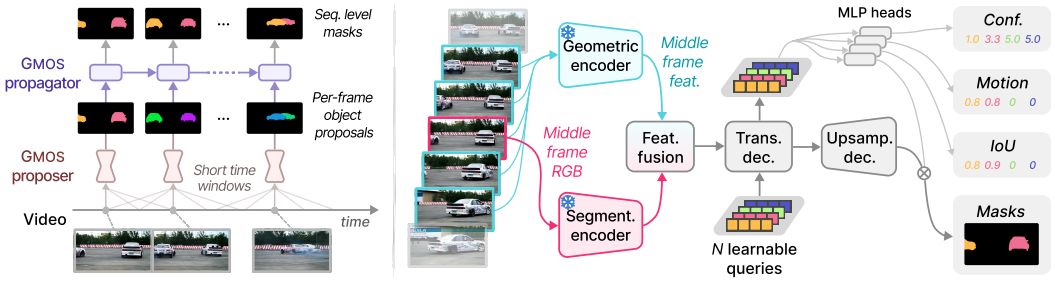

Figure 2: **Overview of GMOS. Left:** The overall *proposer–propagator* design. The proposer operates on a short temporal window (around 0.5 s) and outputs per-frame object proposals, which the propagator links into coherent tracks across the full video. **Right:** The GMOS proposer. A frozen π^3 geometric encoder ingests frames for a short temporal window and a frozen SAM2 segmentation encoder processes the middle frame of the window, to make predictions for the middle frame. Their features are fused and passed through a transformer decoder with N learnable object queries, decoded into segmentation masks for moving objects alongside motion states, estimated IoUs, and confidence scores.

GMOS achieves state-of-the-art results across MOS, MOS-I, and Unsupervised VOS (UVOS) benchmarks, while running roughly three times faster than prior MOS methods (Tab. 1).

2 Method: GMOS

Given an RGB video $\mathcal{V} = \{I_t\}_{t=1}^{T}$, our goal is to predict, for every frame t and every independently moving object i, a segmentation mask together with its motion state,

$$\{(\hat{M}_t^{(i)}, \hat{m}_t^{(i)})\}_{t=1,\dots,T;i=1,\dots,N} = \Phi_{\text{GMOS}}(\mathcal{V}), \tag{1}$$

where $\hat{M}_t^{(i)} \in \{0,1\}^{H\times W}$ is a binary segmentation mask, $\hat{m}_t^{(i)} \in \{0,1\}$ is the motion state with $\hat{m}_t^{(i)} = 1$ indicating that object i is moving at frame t. GMOS produces these outputs jointly, and therefore covers both MOS-I (Sec. 4.2) and the conventional MOS settings.

Fig. 2 (left) provides an overview of GMOS, which follows a *proposer-propagator* design. The proposer operates on a short local window and outputs, for the middle frame, a set of N per-object proposals, each comprising a moving object mask and a predicted motion state. The propagator then links these per-frame proposals into consistent object tracks across the full video.

2.1 GMOS proposer

As illustrated in Fig. 2 (right), the proposer takes raw video frames as its sole input and predicts, for each of N object queries, a segmentation mask together with auxiliary outputs: a confidence score, a motion state, and an estimated mask IoU.

Geometric encoder. To achieve *3D-awareness*, we extract geometric features using a frozen π^3 encoder [72]. Rather than feeding the entire sequence, we process a short temporal window $[t-n, t+n]$ centred at the query frame t, sufficient for distinguishing independent object motion from camera-induced parallax. After π^3 encoding, we retain only the *middle-frame* feature $f_t^{\text{geo}} \in \mathbb{R}^{h_g \times w_g \times d_g}$, as inter-frame correspondence is already encoded in it through the model's cross-frame attention.

Segmentation encoder. Since geometric features alone encode little about object segmentation cues, we leverage the strong objectness prior of a frozen SAM2 encoder [60]. Only the query frame I_t is passed through SAM2, yielding a bottleneck feature $f_t^{\text{seg}} \in \mathbb{R}^{h\times w\times d}$ together with high-resolution features that are later used by the mask head.

Feature fusion. The geometric feature f_t^{geo} is typically at a lower spatial resolution than the segmentation one f_t^{seg}, *i.e.* $h_g < h$. We therefore bilinearly interpolate f_t^{geo} to (h,w), concatenate the two features channel-wise, and apply a linear projection to obtain the fused feature $f_t \in \mathbb{R}^{h\times w\times d}$. A light-weight transformer self-attention encoder is then applied to f_t to enforce global spatial consistency.

Transformer decoder. Following the convention in query-based segmentation [7, 36], we introduce N learnable object queries $\{q^{(i)}\}_{i=1}^{N}$ with $q^{(i)} \in \mathbb{R}^d$ and feed them into a two-way transformer decoder [31]

together with f_t. Each layer alternates query self-attention with query-to-feature and feature-to-query cross-attention, producing refined query tokens $\{q_t^{*(i)}\}$ and refined dense features $f_t^* \in \mathbb{R}^{h\times w\times d}$.

Mask head. The refined dense features f_t^* are upsampled to full resolution through transposed convolutions, with the SAM2 high-resolution features added as skip connections. An MLP maps each refined query $q_t^{*(i)}$ to a mask embedding, and the per-query mask $\hat{M}_t^{(i)}$ is obtained by dot product with the upsampled features, following the hypernetwork design in [7]. During training, queries are matched to ground truth via bipartite Hungarian assignment. Note, an object's ground-truth mask is defined as *empty* whenever the object is not moving, so supervision jointly shapes the mask and the motion state.

Motion and IoU heads. Two independent MLP heads take in the refined query $q_t^{*(i)}$. The *motion head* leads to the probability $\hat{m}_t^{(i)}$ that object i is moving at frame t, and the *IoU head* produces a scalar $\hat{u}_t^{(i)}$ estimating the quality of $\hat{M}_t^{(i)}$ in terms of its expected IoU against the ground truth.

Confidence prediction. The instantaneous motion state is inherently ambiguous near transitions between motion and rest, where the ground truth switches abruptly between a full mask with $m_t^{(i)}=1$ and an empty mask with $m_t^{(i)}=0$. To mitigate this label noise, we follow [70] and introduce a learnable per-query confidence $\hat{c}_t^{(i)} \in [0,1]$, predicted by a final MLP head with sigmoid output. The resulting confidence modulates the mask and motion losses (see Eq. 2).

Training objective. The total proposer loss reads

$$\mathcal{L} = \lambda_{\text{iou}}\mathcal{L}_{\text{iou}} + \tilde{c}\big(\lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{motion}}\mathcal{L}_{\text{motion}}\big) - \big[\lambda_{\text{conf}}^{\text{mov}}\, m + \lambda_{\text{conf}}^{\text{sta}}\,(1-m)\big]\log\tilde{c} \tag{2}$$

The loss is averaged over object queries i and frames t, with the corresponding indices dropped for clarity. Here, $\mathcal{L}_{\text{mask}}$ is a mixture of focal and dice losses, $\mathcal{L}_{\text{motion}}$ is a binary cross-entropy on the motion state, and $\mathcal{L}_{\text{iou}}$ is an MSE loss between the predicted IoU and the actual IoU against the ground truth.

The mask and motion losses are modulated by a rescaled confidence $\tilde{c} = (c_{\max}-1)\hat{c}+1 \in [1, c_{\max}]$, and the last term is an adversarial confidence regulariser [70] that raises $\tilde{c}$ on clean, easy samples the model can solve precisely. For difficult examples, such as a walking pedestrian coming to a stop, the mask and motion losses are large due to label ambiguity around the transition. The model therefore lowers $\tilde{c}$ to attenuate the multiplicative term, paying a small $\log\tilde{c}$ penalty in exchange.

In practice, the confidence would be driven up only for static samples, since predicting an empty mask for a static object is trivially easier than predicting a precise moving-object mask, failing to address the transition noise. We therefore *split* the regularisation by the ground-truth motion state m, using distinct coefficients $\lambda_{\text{conf}}^{\text{mov}}$ and $\lambda_{\text{conf}}^{\text{sta}}$ to calibrate moving and static samples independently.

GMOS-S: *a foreground–background variant.* For efficiency-critical deployment, we additionally instantiate GMOS-S ("S" for *single* binary mask prediction), which predicts all moving objects into a single foreground mask. GMOS-S shares the encoders, feature fusion, and self-attention module with the GMOS proposer, but (i) removes the query-based transformer decoder, (ii) replaces the per-query mask head with a single convolutional decoder that outputs one binary foreground mask per frame, and (iii) dispenses with the follow-up propagator entirely. This makes GMOS-S an end-to-end model that produces instantaneous (per-frame) motion segmentation and achieves lower inference latency than GMOS. Full architectural details are provided in Appendix A.1.

2.2 GMOS propagator

For every frame t, the proposer yields a set of N candidate tuples $\{(\hat{M}_t^{(i)}, \hat{m}_t^{(i)}, \hat{u}_t^{(i)}, \hat{c}_t^{(i)})\}_{i=1}^{N}$. The role of the propagator is to lift these per-frame proposals $\{\hat{M}_t^{(i)}\}$ into coherent object tracks across the full video, guided by the predicted motion probabilities $\{\hat{m}_t^{(i)}\}$ and estimated IoUs $\{\hat{u}_t^{(i)}\}$. The confidence scores $\{\hat{c}_t^{(i)}\}$ are not used here, as they primarily serve as regularisers during proposer training. We build this process on a SAM2 video predictor [60] used as a mask-conditioned tracker, and define an elementary propagation step from which we build online and offline procedures.

Basic propagation step. Given a start frame and a direction, SAM2 propagates the currently tracked masks frame by frame. At each visited frame t, the proposer predictions are first filtered by thresholding on the motion probabilities $\{\hat{m}_t^{(i)}\}$ and estimated IoUs $\{\hat{u}_t^{(i)}\}$. The remaining high-quality moving-object proposals are then matched to the propagated masks via Hungarian assignment based on mask IoU, and the set of active tracks is updated according to one of two cases:

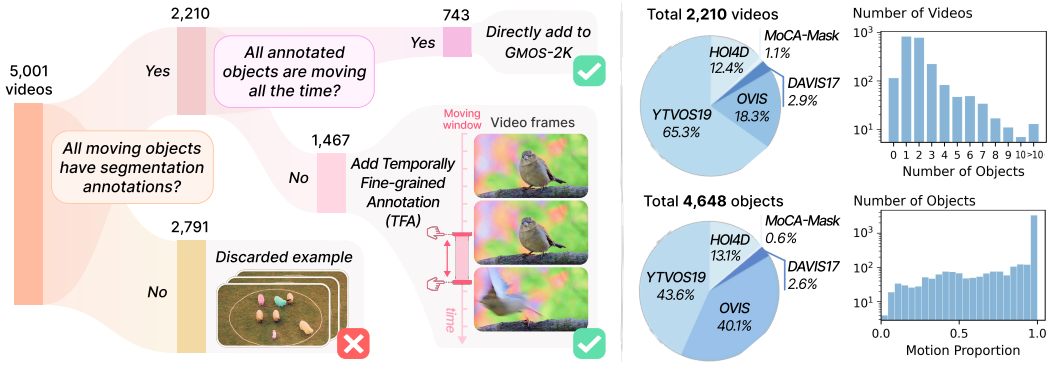

Figure 3: **GMOS-2K overview. Left:** Curation pipeline. 5,001 videos from five VOS datasets are filtered by two criteria, with 743 added directly to GMOS-2K and 1,467 requiring Temporal Fine-grained Annotation (TFA), which labels per-object motion intervals along the time axis. **Right:** Dataset statistics. Pie charts show the per-subset distribution of videos and objects. The upper histogram reports the number of moving objects per video, and the lower one reports the per-object motion proportion, *i.e.* the fraction of the sequence in which each object is actively moving.

- *Addition.* If a proposer prediction matches no existing SAM2 track, it is instantiated as a new mask track starting from frame t.
- *Reinforcement.* If a proposer prediction matches an existing SAM2 track at high IoU, it is injected as an additional prompt for that track, suppressing drift.

For both cases above, the track's motion state at frame t is set to moving. Conversely, if an existing track has a propagated mask at frame t but no proposer prediction falls on the same region (*e.g.* the object was moving earlier but has become static), the track's motion state at frame t is set to static.

Online procedure. The online procedure initialises the tracker from the *first frame* and runs the basic propagation steps in a single causal forward pass. This streams a mask and motion state for each frame as soon as it is processed, supporting MOS-I. Since the sequence is viewed only once and causally, masks cannot be recovered for frames preceding an object's first observed motion, making the online procedure incompatible with the conventional MOS setting that requires a mask at every frame.

Offline procedure. The offline procedure builds on top of the online results with a second propagation stage. It first selects an *anchor frame* $t^\star$ from the online output, defined as the frame whose moving object proposals carry the highest aggregate predicted IoU. The basic propagation step is then run *bi-directionally*, forward from $t^\star$ to T and backward from $t^\star$ to 1, yielding a dense mask for every object at every frame. This mode can be used for both MOS and MOS-I. Further algorithmic details and pseudo-code are provided in Appendix A.2.

3 Dataset: GMOS-2K

We introduce GMOS-2K, a video moving object segmentation dataset with temporally fine-grained motion labels. Beyond ground-truth masks for moving objects, it specifies the exact temporal windows during which each object is in motion, enabling precise training and evaluation of time-sensitive motion segmentation methods.

Curation pipeline. Temporal motion labels are substantially cheaper to obtain than dense segmentation masks. We therefore build upon five existing video object segmentation datasets (DAVIS17 [58], YTVOS19 [78], OVIS [59], MoCA-Mask [11], and HOI4D [45]) and annotate them with per-object motion timestamps. Fig. 3 (left) illustrates the data filtering and annotation pipeline. Starting from 5,001 candidate videos, we apply two sequential filters:

(*i*) *Do all independently moving objects in the video have ground-truth segmentation masks?*
If yes, the sequence proceeds to the second question, otherwise it is filtered out. Note, an object is considered "moving" if it undergoes motion independent of the camera for at least one frame. This question concerns only moving-object masks. Static objects without masks are permissible.

(*ii*) *Are all annotated objects moving throughout the entire sequence?*
If yes, no additional annotation is needed. Otherwise, the video is forwarded for Temporally Fine-grained Annotation (TFA).

Temporally Fine-grained Annotation (TFA). Given a video sequence, TFA aims to annotate time intervals on the temporal axis that specify when objects are moving, with each object annotated independently. The annotation process follows that for video temporal grounding. Further details on the annotation interface, guidelines, and quality controls are provided in Appendix D.

Dataset statistics. GMOS-2K comprises $2{,}210$ videos with $4{,}648$ annotated moving objects, drawn from five constituent subsets that span diverse domains: heavily occluded scenes (OVIS), hand-object interactions (HOI4D), and wildlife and human-centric clips (DAVIS17, YTVOS19, MoCA-Mask). Example sequences are shown in Fig. C2, with instantaneously moving objects labelled.

The pie charts in Fig. 3 (right) show the per-subset distribution of videos and objects. OVIS contributes a disproportionately large share of objects, reflecting its dense multi-object scenes. The accompanying histograms characterise two complementary aspects of the dataset. The per-video object count (top) concentrates on sequences with 0–4 moving objects, with a long tail extending beyond 4. The motion proportion (bottom), defined as the fraction of frames in which an object is actively moving, peaks at fully moving objects, with the remaining cases distributed roughly uniformly across lower values.

The training and test splits consist of $1{,}930$ and 280 videos, respectively. All five subsets contribute to the training split, while DAVIS17 and YTVOS19 additionally contribute to the test split designed for MOS-I evaluation, renamed DAVIS17-IM and YTVOS19-IM ("IM" for Instantaneous Motion) to distinguish them from the conventional MOS datasets. Full per-subset statistics are reported in Tab. C1.

4 Experiments

4.1 Training datasets

We train GMOS on a composite dataset of $14{,}171$ videos (782K annotated frames), combining three data sources. (*i*) *Synthetic data.* Kubric [22], PointOdyssey [89], and DynamicReplica [27] provide $10{,}526$ videos with pixel-perfect ground truth and motion labels. (*ii*) *Real moving-object data (*GMOS*-2K).* The five GMOS-2K training subsets (Sec. 3) contribute $1{,}930$ real-world videos with temporally fine-grained motion labels. (*iii*) *Real static-scene data.* $1{,}715$ clips from the Mannequin Challenge dataset [39], in which people remain frozen while the camera moves freely, provide static-scene supervision that teaches the model to distinguish depth parallax from independent object motion. The detailed per-dataset statistics are provided in Tab. C1.

4.2 Benchmarks for evaluation

We evaluate GMOS on the following video object segmentation tasks.

Moving Object Segmentation (MOS) predicts, at every frame, a segmentation mask for every object that moves at any point in the sequence, regardless of its per-frame motion state. We evaluate on established MOS benchmarks: DAVIS16-M [24] ("M" for "Moving"), DAVIS17-M [15], SegTrackv2 [37], FBMS-59 [51], and MoCA [32], along with YTVOS19-M, a new benchmark derived from the GMOS-2K test set. Following standard practice, all benchmarks report region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$, except MoCA, which uses detection success rate (SR) [79].

MOS-I ("I" for *instantaneous*) is a temporally fine-grained evaluation protocol for MOS that (i) retains the sequence-level object identity association of MOS, (ii) scores segmentation masks only on frames where an object is actively moving, and (iii) explicitly penalises false-positive predictions on static or background objects. We instantiate it on two new benchmarks derived from the GMOS-2K test set, namely DAVIS17-IM and YTVOS19-IM ("IM" for "Instantaneous Motion"), and report performance on three complementary metrics.

- *Moving-object Jaccard* ($\mathcal{J}_{\text{mov}}$) – segmentation accuracy on actively moving frames. For each annotated object, $\mathcal{J}$ is averaged over the frames in which it is labelled as moving in GMOS-2K.
- *False-positive count* (FP count) – penalisation of static and background predictions. We measure the average number of static or background objects per frame that are incorrectly predicted as moving. While $\mathcal{J}_{\text{mov}}$ rewards detecting true motion, FP count penalises spurious motion claims.
- *Mean temporal IoU* (mtIoU) – joint measure of moving-object detection and false-positive suppression. For each sequence, we compute a temporal IoU between predicted and ground-truth moving objects, counted *per object per frame*. The *union* covers all (object, frame) entries marked as moving by either the prediction or the ground truth. The *intersection* covers entries where a

Table 2: **Encoders in the GMOS proposer.** π^3 and SAM2 are adopted as our default encoders. We report the proposer's performance directly, with the propagator skipped.

Geometric encoder	Segmentation encoder	YTVOS19-IM			DAVIS17-IM		
		mtIoU↑	$\mathcal{J}_{\text{mov}}$ ↑	FP count↓	mtIoU↑	$\mathcal{J}_{\text{mov}}$ ↑	FP count↓
✗	SAM2 [60]	49.9	67.0	0.279	57.9	74.8	0.174
VGGT [69]	SAM2 [60]	60.5	75.0	0.194	67.4	80.4	0.112
DA3 [42]	SAM2 [60]	61.3	75.2	0.157	67.7	80.6	0.141
π^3 [72]	✗	59.8	73.1	**0.144**	62.1	77.1	**0.075**
π^3 [72]	SAM3 [3]	**62.9**	75.8	0.155	68.7	**82.5**	0.157
RAFT [65]	SAM2 [60]	56.1	72.7	0.229	63.8	79.6	0.167
π^3 [72] + RAFT [65]	SAM2 [60]	61.7	75.1	0.153	**69.2**	82.3	0.135
π^3 [72]	SAM2 [60]	62.1	**76.6**	0.175	68.7	82.2	0.147

predicted and a ground-truth moving object are matched by region similarity $\mathcal{J}$ at threshold k. Sweeping $k \in \{0.5, 0.55, ..., 0.95\}$ and averaging the resulting tIoUs yields the final mtIoU.

Unsupervised Video Object Segmentation (UVOS) requires automatically segmenting salient objects in a video, where targets are defined by perceptual salience and need not be moving. We report results on DAVIS16 [57] and DAVIS17 [58] using the standard region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$.

4.3 Implementation details

We elaborate the design choices of GMOS below, with further details provided in Appendix B.

Architecture. GMOS uses pretrained π^3 [72] (DINOv2 [54] ViT-L/14) and SAM2 [60] (Hiera-L [62]) as the geometric and segmentation encoders, both kept frozen. Bottleneck features have spatial resolution 64×64 and channel dimension 256. The trainable transformer encoder and decoder each have 3 layers and 8 attention heads. We use 100 object queries and predict masks at 256×256.

Input setting. The GMOS proposer operates on 5 frames sampled uniformly from a temporal window centred on the target frame. The window size is randomly sampled between 0.16 s and 1.00 s during training to expose the model to a range of motion speeds, and fixed at 0.5 s at inference. As the window is centred, the proposer requires two future frames at inference, introducing a fixed lookahead of 0.25 s.

Training setup. We optimise with AdamW [46] using learning rate 1×10^{-4}, weight decay 0.01, gradient clipping at norm 1.0, and a cosine schedule with $2{,}000$ warmup steps. Training runs on 2 NVIDIA RTX A6000 GPUs for approximately 2 days, with an effective batch size of 8. For the proposer loss in Eq. 2, we set $(\lambda_{\text{mask}}, \lambda_{\text{motion}}, \lambda_{\text{iou}}, \lambda_{\text{conf}}^{\text{mov}}, \lambda_{\text{conf}}^{\text{sta}}) = (20, 1, 1, 0.1, 0.001)$, with a rescaled confidence upper bound $c_{\max} = 5$. The mask loss combines focal (α=0.25, γ=2) and dice losses in a 20:1 ratio.

4.4 Ablation study

We validate the encoder design of the GMOS proposer. To probe the proposer directly, we skip the propagator and apply per-frame Hungarian matching between proposer outputs and the ground truth, reporting performance on YTVOS19-IM and DAVIS17-IM.

Tab. 2 ablates the encoder choices for the GMOS proposer. Removing the geometric encoder destroys the model's grounding ability, producing many hallucinated false positives. Replacing π^3 with VGGT or DA3 yields a modest drop, suggesting that π^3 provides richer 4D priors that better disentangle object motion from camera motion.

We further compare π^3 against an optical-flow-based motion encoder under identical training data. Following GeoMotion [23], we extract forward and backward optical flow with RAFT [65] and encode it with a trainable CNN. The resulting flow features replace, or are added to, the π^3 features and are fused with the SAM2 features as in Sec. 2.1. As a single motion source, π^3 clearly outperforms optical flow on both benchmarks, while adding optical flow on top of π^3 changes the results only marginally.

Dropping the segmentation encoder, on the other hand, reduces the false-positive count but substantially degrades mask quality, causing large drops in mtIoU and $\mathcal{J}_{\text{mov}}$. Replacing SAM2 with the more recent SAM3 [3] yields comparable mask quality and only a marginal $\mathcal{J}_{\text{mov}}$ gain. We therefore retain SAM2

Table 3: **Comparison with Moving Object Segmentation (MOS) baselines.** The Δ vs. prev. SoTA row reports the difference between our best result (GMOS or GMOS-S) and the previous state of the art.

Methods	Multi-Object				Foreground–Background										
	YTVOS19-M		DAVIS17-M		YTVOS19-M		DAVIS17-M		DAVIS16-M		STv2		FBMS		MoCA
	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	SR↑
OCLR-TTA [74]	-	-	48.4	49.9	-	-	76.0	75.3	80.1	76.9	72.1	75.3	69.9	68.3	0.559
ABR [75]	-	-	50.9	51.2	-	-	74.6	75.2	70.2	73.7	76.3	81.1	81.9	79.6	-
FlowSAM [76]	66.6	68.2	74.6	81.0	52.4	54.3	87.9	87.0	85.7	83.8	77.9	83.9	82.8	80.9	0.643
SegAnyMo [24]	64.6	66.2	77.4	83.6	70.9	73.0	90.0	89.0	89.2	89.7	76.3	85.4	78.3	82.8	0.675
GeoMotion [23]	-	-	-	-	65.4	65.7	82.2	82.3	83.5	84.3	77.3	84.3	72.5	78.5	0.615
GMOS	**79.9**	**82.2**	**79.7**	**86.0**	**85.1**	**84.6**	90.0	90.6	88.7	90.3	**82.0**	**88.3**	**91.9**	**93.0**	**0.837**
GMOS-S	-	-	-	-	81.4	81.8	**90.7**	**93.1**	**90.0**	**93.2**	78.9	85.9	83.0	84.9	0.785
Δ vs. prev. SoTA	+13.3	+14.0	+2.3	+2.4	+14.2	+11.6	+0.7	+4.1	+0.8	+3.5	+4.1	+2.9	+9.1	+10.2	+0.162

Table 4: **Comparison with MOS-I baselines.** "I" is short for "instantaneous".

Methods	YTVOS19-IM			DAVIS17-IM		
	mtIoU↑	$\mathcal{J}_{\text{mov}}\uparrow$	FP count↓	mtIoU↑	$\mathcal{J}_{\text{mov}}\uparrow$	FP count↓
OCLR-TTA [74]	-	-	-	28.3	54.1	0.550
ABR [75]	-	-	-	36.0	57.0	0.271
FlowSAM [76]	21.1	62.8	3.085	22.7	75.3	4.101
SegAnyMo [24]	55.9	63.2	0.320	67.0	81.7	0.253
GMOS (*online*)	**64.2**	76.4	**0.197**	**72.9**	83.5	**0.075**
GMOS	63.4	**76.6**	0.238	**72.9**	**84.2**	0.079
Δ vs. prev. SoTA	+8.3	+13.4	+0.123	+5.9	+2.5	+0.178

Table 5: **Comparison with Unsupervised VOS (UVOS) baselines.**

Methods	DAVIS17		DAVIS16	
	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
OCLR-TTA [74]	43.7	44.5	80.8	76.8
ABR [75]	45.0	45.3	71.8	73.2
DEVA [9]	70.4	76.4	87.6	90.2
FlowSAM [76]	71.2	76.3	87.1	84.9
SegAnyMo [24]	66.5	71.4	**90.6**	91.0
GeoMotion [23]	-	-	84.5	85.0
GMOS	**76.2**	**80.6**	90.1	**91.5**
GMOS-S	-	-	87.9	90.8
Δ vs. prev. SoTA	+5.0	+4.2	−0.5	+0.5

as the default segmentation encoder for direct comparison with prior methods that also build on it. These trends are consistent across both YTVOS19-IM and DAVIS17-IM, further supporting our choice of π^3 and SAM2 features.

Appendix E provides further ablations on the GMOS proposer (training data composition, confidence-based loss, input temporal window and frame gap, early-layer π^3 features, and mask decoder) and on the GMOS propagator, together with repeated runs across random seeds. It also reports additional analyses, showing that GMOS retains a clear margin when the baselines are equipped with a per-frame motion-state classifier trained on our annotations (Tab. E12), that its margin over prior methods widens under large camera motion (Tab. E13), and that a strictly causal variant of the online procedure remains competitive (Tab. E11).

4.5 Quantitative results

Tab. 3 reports results on the conventional MOS setting, where moving-object masks are predicted at the sequence level. GMOS, and its foreground–background variant GMOS-S, achieves new state-of-the-art performance across all benchmarks. Gains are most pronounced on YTVOS19-M, whose denser and more varied object interactions expose the limitations of prior flow- and trajectory-based methods. GMOS also remains robust on camouflaged objects (MoCA), where segmentation must rely exclusively on motion cues.

As shown in Tab. 4, performance gaps widen further on MOS-I, which requires predictions at finer temporal granularity. Since prior methods treat motion as a sequence-level attribute and cannot capture instantaneous motion states, GMOS surpasses them by a substantial margin on all three metrics. We additionally evaluate an online variant of GMOS (Sec. 2.2), which performs on par with the default offline mode. This can be attributed to the instantaneous nature of MOS-I, where a short local window already provides sufficient motion evidence. The offline mode's second global propagation even slightly hurts FP count and mtIoU, as the bidirectional pass amplifies any borderline false positives from the first pass. The online mode, however, is restricted to the MOS-I setup, since a single forward pass cannot recover masks for objects that are initially static and only later begin to move. The offline procedure therefore remains our default across all tasks, including MOS and UVOS.

We also report performance on Unsupervised Video Object Segmentation (UVOS), which targets the prediction of salient objects regardless of their motion state. Although UVOS lies outside our primary focus of segmenting only moving objects, Tab. 5 shows that GMOS generalises well to this adjacent

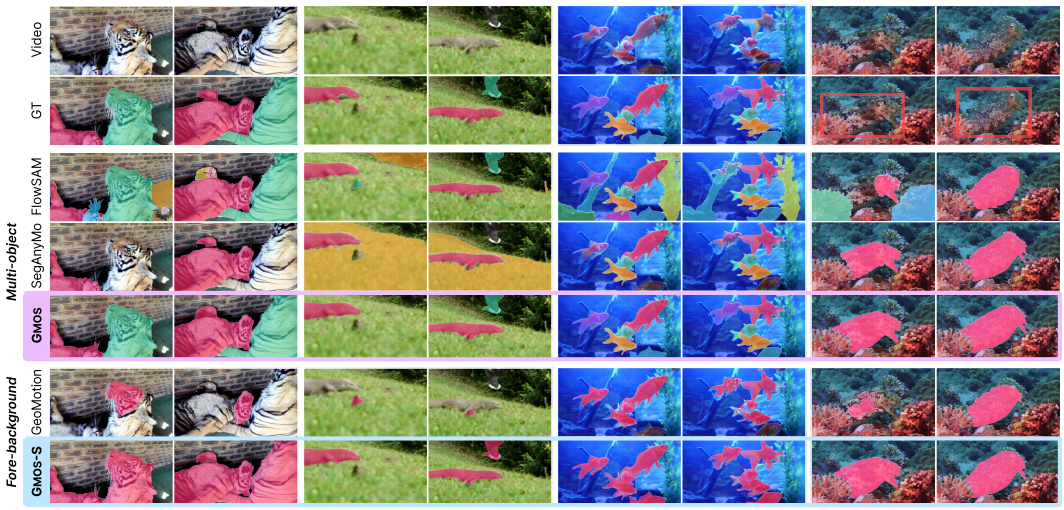

Figure 4: **Qualitative comparison on the MOS task.** Example videos are sampled from YTVOS19 (first two columns), DAVIS17 (third column), and MoCA (last column). The middle block shows multi-object results, and the bottom block shows foreground–background results.

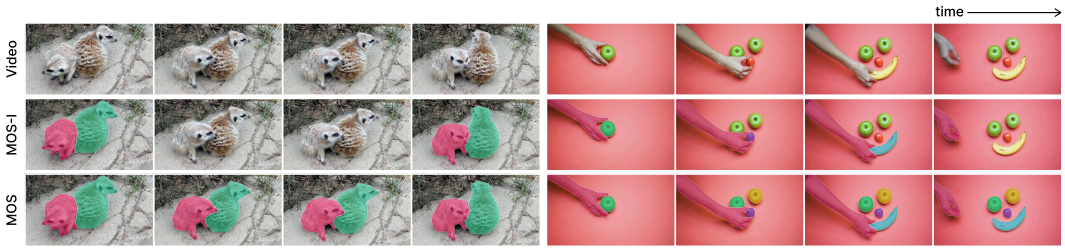

Figure 5: **GMOS on in-the-wild videos.** Two in-the-wild sequences (sourced outside our training or test datasets) illustrate our MOS and MOS-I predictions. Under MOS-I, instantaneously moving objects are segmented, with object identities consistently associated across frames. Under MOS, GMOS produces full segmentation masks for every object that moves at any point in the sequence, regardless of its motion state at a given frame.

task, achieving state-of-the-art results on the multi-object DAVIS17 benchmark and remaining on par with the prior best on the single-object DAVIS16.

4.6 Qualitative results

Fig. 4 compares GMOS against representative baselines on sequences from YTVOS19, DAVIS17, and MoCA. Regarding multi-object predictions, FlowSAM [76] tends to over-segment moving objects (*e.g.* treating the coral as moving), while SegAnyMo [24] misses moving objects or bleeds masks into the background. In contrast, GMOS assigns each moving object a consistent identity and boundary that closely follow the ground truth, even under heavy occlusion. Under the foreground–background protocol, GeoMotion [23] captures prominent motion but overlooks instances near image boundaries (*e.g.* the right tiger), whereas GMOS-S recovers them reliably.

Fig. 5 illustrates GMOS on in-the-wild videos, contrasting MOS-I and MOS predictions. In the left sequence, the meerkats become static mid-sequence. Our MOS-I predictions correctly identify the static states, while MOS produces consistent segmentation across frames. In the right sequence, a hand arranges fruits one piece at a time. Our MOS-I predictions correctly release each piece once it comes to rest, whereas MOS keeps them masked throughout.

5 Related works

Unsupervised Video Object Segmentation (UVOS) aims to identify salient objects in videos without any annotation at inference time, in contrast to semi-supervised VOS [8, 10, 53, 57, 58, 60, 78, 84],

which propagates masks from first-frame ground truths. Early methods adopt two-stream architectures that fuse motion and appearance inputs [25, 26, 43, 47, 56, 61, 80, 87, 90], while others rely primarily on saliency or attention mechanisms [12, 13, 19, 34, 35, 44, 64, 68, 83, 85, 91]. More recent works scale UVOS with new data strategies and modular pipeline designs: VideoCutLER [71] scales training via copy-paste augmentation transferred from unsupervised image instance segmentation, DEVA [9] composes an image-level segmenter with a separate temporal propagator, and SAM-based approaches [3, 31, 60, 88] build UVOS pipelines around pre-trained foundation segmenters. Despite significant progress, UVOS methods typically target the most salient objects, which do not necessarily correspond to objects that are actually moving.

Motion Segmentation (MS) seeks to separate independently moving regions from the static background, producing a binary foreground–background mask. Mainstream approaches take optical flow as the motion signal and cluster pixels according to similar motion patterns [1, 14, 15, 32, 48, 49, 55, 66, 67, 79, 81, 82]. However, these flow-based approaches are inherently bound by the quality of the pre-computed optical flow, and also vulnerable to large camera motion and induced depth parallax. A complementary line of work aggregates motion cues over longer temporal windows via point trajectories [2, 20, 30, 52]. Recent advances in point tracking [16, 17, 28] have substantially improved long-range correspondence, enabling segmentation to be learnt directly from dense trajectories [24, 29]. However, such reliance on long-term tracking makes these methods insensitive to static–dynamic transitions, limiting their online applicability.

Another line of work exploits 3D or 4D geometric priors for motion segmentation. MegaSAM [40] learns motion probability maps from estimated camera parameters and depth to separate moving and static content, while RoMo [21] combines optical flow with epipolar constraints and iteratively refines motion masks using segmentation priors from SAM [31]. Building upon feed-forward reconstruction methods [69, 70], MonST3R [86] and DAS3R [77] jointly decode moving regions alongside temporally consistent video reconstructions. Easi3R [6] reveals that the attention layers of such reconstruction models already encode relative camera-object motion. GeoMotion [23] further exploits the latent 4D features of π^3 [72] to directly disentangle object and camera motion in a feed-forward manner.

Moving Object Segmentation (MOS) aims to discover, segment, and track individual moving objects. For videos containing a single moving object, MOS degenerates to foreground–background motion segmentation. Recent works instead focus on the more challenging multi-object scenario, similarly building upon the aforementioned motion cues (*i.e.* flow, trajectories, and geometric priors).

Without relying on human annotations, one group of methods [33, 41, 63] pursues purely unsupervised training via flow reconstruction as a proxy objective. Alternatively, OCLR [74] takes optical flow as input and trains object-centric layered representations on synthetic data to discover and track multiple objects under occlusion, while ABR [75] extends it with appearance-based refinement.

With increasingly available annotated masks and strong pre-trained models, FlowSAM [76] injects optical flow into SAM [31], either as an input channel or as prompt signals, for moving object segmentation. SegAnyMo [24] fuses long-range trajectory cues with DINO [4] features and employs SAM2 [60] for mask densification.

6 Conclusion

We presented GMOS, a feed-forward framework that grounds Moving Object Segmentation in 3D space and time, making it robust to camera-induced parallax in space and temporally sensitive to per-frame motion-state changes. GMOS operates on RGB video alone and supports both online and offline inference. To facilitate training and evaluation in this regime, we curated GMOS-2K, a real-world video dataset with per-object temporal motion annotations, and formalised MOS-I, a temporally fine-grained evaluation protocol with three complementary metrics. GMOS achieves state-of-the-art results on MOS, MOS-I, and UVOS benchmarks, while running roughly three times faster than prior MOS methods.

Acknowledgements. This research is supported by the UK EPSRC Programme Grant VisualAI (EP/T028572/1), the Royal Society Research Professorship RSRP\R\241003, and the Clarendon Scholarship.

References

[1] Pia Bideau and Erik Learned-Miller. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In *ECCV*, 2016.

[2] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010.

[3] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, RISHI HAZRA, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In *ICLR*, 2026.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[5] Xieyuanli Chen, Shijie Li, Benedikt Mersch, Louis Wiesmann, Jürgen Gall, Jens Behley, and Cyrill Stachniss. Moving object segmentation in 3D LiDAR data: A learning-based approach exploiting sequential data. In *IROS*, 2021.

[6] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3r: Estimating disentangled motion from dust3r without training. In *ICCV*, 2025.

[7] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021.

[8] Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022.

[9] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *ICCV*, 2023.

[10] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *CVPR*, 2024.

[11] Xuelian Cheng, Huan Xiong, Deng-Ping Fan, Yiran Zhong, Mehrtash Harandi, Tom Drummond, and Zongyuan Ge. Implicit motion handling for video camouflaged object detection. In *CVPR*, 2022.

[12] S. Cho, M. Lee, S. Lee, C. Park, D. Kim, and S. Lee. Treating motion as option to reduce motion dependency in unsupervised video object segmentation. In *WACV*, 2023.

[13] Suhwan Cho, Minhyeok Lee, Seunghoon Lee, Dogyoon Lee, Heeseung Choi, Ig-Jae Kim, and Sangyoun Lee. Dual prototype attention for unsupervised video object segmentation. In *CVPR*, 2024.

[14] Subhabrata Choudhury, Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion. In *BMVC*, 2022.

[15] Achal Dave, Pavel Tokmakov, and Deva Ramanan. Towards segmenting anything that moves. In *ICCV*, 2019.

[16] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. In *NeurIPS*, 2022.

[17] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *ICCV*, 2023.

[18] Abhishek Dutta and Andrew Zisserman. The via annotation software for images, audio and video. In *ACM MM*, 2019.

[19] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *CVPR*, 2019.

[20] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, 2012.

[21] Lily Goli, Sara Sabour, Mark Matthews, Brubaker Marcus, Dmitry Lagun, Alec Jacobson, David J. Fleet, Saurabh Saxena, and Andrea Tagliasacchi. RoMo: Robust motion segmentation improves structure from motion. In *ICCV*, 2025.

[22] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator. In *CVPR*, 2022.

[23] Xiankang He, Peile Lin, Ying Cui, Dongyan Guo, Chunhua Shen, and Xiaoqin Zhang. Geomotion: Rethinking motion segmentation via latent 4d geometry. In *CVPR*, 2026.

[24] Nan Huang, Wenzhao Zheng, Chenfeng Xu, Kurt Keutzer, Shanghang Zhang, Angjoo Kanazawa, and Qianqian Wang. Segment any motion in videos. In *CVPR*, 2025.

[25] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, 2017.

[26] Ge-Peng Ji, Keren Fu, Zhe Wu, Deng-Ping Fan, Jianbing Shen, and Ling Shao. Full-duplex strategy for video object segmentation. In *ICCV*, 2021.

[27] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *CVPR*, 2023.

[28] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labeling real videos. In *ICCV*, 2025.

[29] Laurynas Karazija, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Learning segmentation from point trajectories. In *NeurIPS*, 2024.

[30] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *ICCV*, 2015.

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *ICCV*, 2023.

[32] Hala Lamdouar, Charig Yang, Weidi Xie, and Andrew Zisserman. Betrayed by motion: Camouflaged object discovery via motion segmentation. In *ACCV*, 2020.

[33] Dong Lao, Zhengyang Hu, Francesco Locatello, Yanchao Yang, and Stefano Soatto. Divided attention: Unsupervised multi-object discovery with contextually separated slots. *arXiv preprint arxiv:2304.01430*, 2023.

[34] M. Lee, S. Cho, S. Lee, C. Park, and S. Lee. Unsupervised video object segmentation via prototype memory network. In *WACV*, 2023.

[35] Minhyeok Lee, Suhwan Cho, Dogyoon Lee, Chaewon Park, Jungho Lee, and Sangyoun Lee. Guided slot attention for unsupervised video object segmentation. In *CVPR*, 2024.

[36] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *CVPR*, 2023.

[37] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013.

[38] Qipeng Li, Yuan Zhuang, You Chen, Jianzhu Huai, Miaomiao Li, Tianxiang Ma, Yufei Tang, and Xinlian Liang. Multi-sensor fusion for robust localization with moving object segmentation in complex dynamic 3D scenes. *International Journal of Applied Earth Observation and Geoinformation*, 2023.

[39] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T. Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, 2019.

[40] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *CVPR*, 2025.

[41] Long Lian, Zhirong Wu, and Stella X. Yu. Bootstrapping objectness from videos by relaxed common fate and visual grouping. In *CVPR*, 2023.

[42] Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *ICLR*, 2026.

[43] Daizong Liu, Dongdong Yu, Changhu Wang, and Pan Zhou. F2net: Learning to focus on the foreground for unsupervised video object segmentation. In *AAAI*, 2021.

[44] Weihuang Liu, Xi Shen, Haolun Li, Xiuli Bi, Bo Liu, Chi-Man Pun, and Xiaodong Cun. Depth-aware test-time training for zero-shot video object segmentation. In *CVPR*, 2024.

[45] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *CVPR*, 2022.

[46] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[47] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *CVPR*, 2019.

[48] Etienne Meunier and Patrick Bouthemy. Unsupervised motion segmentation in one go: Smooth long-term model over a video. *arXiv preprint arXiv:2310.01040*, 2023.

[49] Etienne Meunier, Anaïs Badoual, and Patrick Bouthemy. Em-driven unsupervised learning for efficient motion segmentation. *IEEE TPAMI*, 2022.

[50] Nasaruddin Nasaruddin, Kahlil Muchtar, Afdhal Afdhal, and Alvin Prayuda Juniarta Dwiyantoro. Deep anomaly detection through visual attention in surveillance videos. *Journal of Big Data*, 2020.

[51] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE TPAMI*, 2014.

[52] Peter Ochs and Thomas Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011.

[53] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[54] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024.

[55] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013.

[56] Gensheng Pei, Fumin Shen, Yazhou Yao, Guo-Sen Xie, Zhenmin Tang, and Jinhui Tang. Hierarchical feature alignment network for unsupervised video object segmentation. In *ECCV*, 2022.

[57] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[58] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[59] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip Torr, and Song Bai. Occluded video instance segmentation: A benchmark. *IJCV*, 2022.

[60] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[61] Sucheng Ren, Wenxi Liu, Yongtuo Liu, Haoxin Chen, Guoqiang Han, and Shengfeng He. Reciprocal transformations for unsupervised video object segmentation. In *CVPR*, 2021.

[62] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, Jitendra Malik, Yanghao Li, and Christoph Feichtenhofer. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *ICML*, 2023.

[63] Sadra Safadoust and Fatma Güney. Multi-object discovery by low-dimensional object motion. In *ICCV*, 2023.

[64] Huihui Song, Tiankang Su, Yuhui Zheng, Kaihua Zhang, Bo Liu, and Dong Liu. Generalizable fourier augmentation for unsupervised video object segmentation. In *AAAI*, 2024.

[65] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[66] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017.

[67] Pavel Tokmakov, Cordelia Schmid, and Karteek Alahari. Learning to segment moving objects. *IJCV*, 2019.

[68] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *CVPR*, 2019.

[69] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025.

[70] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024.

[71] Xudong Wang, Ishan Misra, Zizun Zeng, Rohit Girdhar, and Trevor Darrell. Videocutler: Surprisingly simple unsupervised video instance segmentation. *arXiv preprint arXiv:2308.14710*, 2023.

[72] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *ICLR*, 2026.

[73] Zhengkui Weng, Zhipeng Jin, Shuangxi Chen, Quanquan Shen, Xiangyang Ren, and Wuzhao Li. Attention-based temporal encoding network with background-independent motion mask for action recognition. *Computational Intelligence and Neuroscience*, 2021.

[74] Junyu Xie, Weidi Xie, and Andrew Zisserman. Segmenting moving objects via an object-centric layered representation. In *NeurIPS*, 2022.

[75] Junyu Xie, Weidi Xie, and Andrew Zisserman. Appearance-based refinement for object-centric motion segmentation. In *ECCV*, 2024.

[76] Junyu Xie, Charig Yang, Weidi Xie, and Andrew Zisserman. Moving object segmentation: All you need is sam (and flow). In *ACCV*, 2024.

[77] Kai Xu, Tze Ho Elden Tse, Jizong Peng, and Angela Yao. Das3r: Dynamics-aware gaussian splatting for static scene reconstruction. *arXiv preprint arxiv:2412.19584*, 2024.

[78] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. In *ECCV*, 2018.

[79] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *ICCV*, 2021.

[80] S. Yang, L. Zhang, J. Qi, H. Lu, S. Wang, and X. Zhang. Learning motion-appearance co-attention for zero-shot video object segmentation. In *ICCV*, 2021.

[81] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *CVPR*, 2019.

[82] Yanchao Yang, Brian Lai, and Stefano Soatto. Dystab: Unsupervised object segmentation via dynamic-static bootstrapping. In *CVPR*, 2021.

[83] Zhao Yang, Qiang Wang, Luca Bertinetto, Song Bai, Weiming Hu, and Philip H.S. Torr. Anchor diffusion for unsupervised video object segmentation. In *ICCV*, 2019.

[84] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *NeurIPS*, 2022.

[85] Yichen Yuan, Yifan Wang, Lijun Wang, Xiaoqi Zhao, Huchuan Lu, Yu Wang, Weibo Su, and Lei Zhang. Isomer: Isomerous transformer for zero-shot video object segmentation. In *ICCV*, 2023.

[86] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3r: A simple approach for estimating geometry in the presence of motion. In *ICLR*, 2025.

[87] K. Zhang, Z. Zhao, D. Liu, Q. Liu, and B. Liu. Deep transport network for unsupervised video object segmentation. In *ICCV*, 2021.

[88] Zhenghao Zhang, Shengfan Zhang, Zhichao Wei, Zuozhuo Dai, and Siyu Zhu. Uvosam: A mask-free paradigm for unsupervised video object segmentation via segment anything model. *arXiv preprint arXiv:2305.12659*, 2024.

[89] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023.

[90] Tianfei Zhou, Shunzhou Wang, Yi Zhou, Yazhou Yao, Jianwu Li, and Ling Shao. Motion-attentive transition for zero-shot video object segmentation. In *AAAI*, 2020.

[91] Tianfei Zhou, Jianwu Li, Xueyi Li, and Ling Shao. Target-aware object discovery and association for unsupervised video multi-object segmentation. In *CVPR*, 2021.

Appendix

This appendix is organised as follows:

- **Additional method details:** Appendix A describes the architecture and training objective of the GMOS-S variant, and provides the full pseudo-code and elementary-step walkthrough for the GMOS propagator.
- **Additional implementation details:** Appendix B reports the GMOS-S training setup, the propagator thresholds, and the protocol for runtime measurement.
- **Dataset details:** Appendix C provides per-dataset statistics for our training and evaluation sources, along with example visualisations from GMOS-2K.
- **GMOS-2K curation details:** Appendix D details the annotator instructions, the annotation tool, and per-subset curation statistics together with the annotation cost.
- **Further ablation study:** Appendix E presents comprehensive ablations on both the GMOS proposer and the GMOS propagator, alongside repeated training runs across random seeds to assess statistical stability, and further analyses on strictly causal inference, baselines equipped with a per-frame motion-state classifier, and robustness to camera motion.
- **Additional qualitative visualisations:** Appendix F provides further qualitative comparisons with baselines and additional in-the-wild MOS/MOS-I results.
- **Discussion:** Appendix G discusses current failure cases, outlines possible extensions, and addresses broader societal impact.

We additionally release, in the supplementary material, (i) a *visualisation video* covering the evaluation protocol comparison (MOS vs. MOS-I), dataset samples, and model outputs, and (ii) a *code repository* with training and evaluation pipelines, along with temporally fine-grained motion state labels for the GMOS-2K subsets, indexed by dataset, split, and sequence name.

A Additional method details

A.1 GMOS-S

Architecture. GMOS-S is designed as a streaming foreground–background variant of the GMOS proposer. As illustrated in Fig. A1, it shares the frozen geometric encoder (π^3), the frozen segmentation encoder (SAM2), and the feature fusion module with the GMOS proposer (Fig. 2). Two design choices distinguish it from the full proposer: (*i*) the transformer decoder and the N learnable object queries are removed, and the fused feature is passed directly through an upsampling decoder that predicts a single binary foreground–background mask per frame; and (*ii*) since motion is implicit in the union of moving objects, the motion head is dropped, and the remaining confidence and IoU heads operate on globally average-pooled fused features, producing one scalar per frame instead of one per object.

In addition, we dispense with the propagator, as no object-level proposals are produced by GMOS-S. Predictions are therefore instantaneous and online, making GMOS-S the fastest variant in our framework.

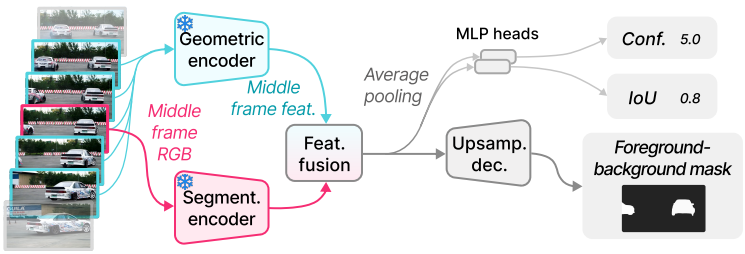

Figure A1: **Overview of GMOS-S,** a streaming foreground–background variant of the GMOS proposer. Instead of query-based decoding, GMOS-S directly upsamples dense features to predict a binary mask.

Training objective. GMOS-S predicts a single frame-level foreground–background mask $\hat{M}_t$, so the per-object Hungarian matching and motion loss used in the GMOS proposer are no longer needed. We therefore simplify Eq. 2 into

$$\mathcal{L}^{\mathrm{S}} = \lambda^{\mathrm{S}}_{\mathrm{iou}}\mathcal{L}^{\mathrm{S}}_{\mathrm{iou}} + \tilde{c}\lambda^{\mathrm{S}}_{\mathrm{mask}}\mathcal{L}^{\mathrm{S}}_{\mathrm{mask}} - \lambda^{\mathrm{S}}_{\mathrm{conf}}\log\tilde{c}, \tag{A1}$$

where $\mathcal{L}^{\mathrm{S}}_{\mathrm{mask}}$ and $\mathcal{L}^{\mathrm{S}}_{\mathrm{iou}}$ are the frame-level counterparts of the proposer's mask and IoU losses in Eq. 2, and $\tilde{c} \in [1, c_{\max}]$ is the rescaled frame-level confidence. Since GMOS-S reasons at the frame level rather than the object level, the split confidence regulariser of the GMOS proposer collapses to a single term.

A.2 GMOS propagator

We now expand on the elementary propagation step and the online/offline procedures outlined in Sec. 2.2 and provide the full pseudo-code in Algorithm 1. The propagator maintains a set of prompts in the SAM2 state $\mathcal{S}$ and produces per-frame masks $\tilde{M}_t$ and motion labels $\tilde{m}_t$.

PROPSTEP. Starting from t_{start} and moving in the given direction, for each frame t:

(*i*) SAM2 propagates the currently tracked masks to frame t, yielding $\tilde{M}_t$.

(*ii*) The proposer runs on a short local window to obtain predictions $\{\hat{M}_t, \hat{m}_t, \hat{u}_t\}$ (*i.e.* mask proposals, motion probabilities, and estimated IoUs), which are Hungarian-matched to $\tilde{M}_t$ on mask IoU.

(*iii*) If update_prompt is enabled, the prompt set is updated by one of two rules applied only to confident moving proposals ($\hat{m}_t^{(i)} > \tau_m$, $\hat{u}_t^{(i)} > \tau_u$): (i) *addition*, when the proposer mask is not covered by any existing track ($\max_j \textsc{Precision}(\hat{M}_t^{(i)}, \tilde{M}_t^{(j)}) < \tau_{\mathrm{new}}$), seeds a new track; (ii) *reinforcement*, when the matched proposer mask aligns with the propagated mask ($\textsc{IoU}(\hat{M}_t^{(i)}, \tilde{M}_t^{(i)}) > \tau_{\mathrm{match}}$), injects $\hat{M}_t^{(i)}$ as an additional prompt for the same track.

(*iv*) The per-frame motion label $\tilde{m}_t^{(i)}$ is inherited from the matched proposer prediction, $\tilde{m}_t^{(i)} = \mathbf{1}[\hat{m}_t^{(i)} > \tau_m]$. Existing tracks for which no confident moving proposer prediction matches at frame t (either unmatched or matched with $\hat{m}_t^{(i)} \le \tau_m$) are therefore labelled static, capturing the "moving-then-static" transition.

Online procedure. We initialise $\mathcal{S}$ with confident moving proposals from frame 1 (*i.e.* the first frame) and run PROPSTEP once in the forward direction with update_prompt = True. The procedure traverses the sequence once in temporal order and streams per-frame outputs as the video arrives, making it directly suitable for MOS-I deployment.

Offline procedure. We first run the online procedure, then select the top-K highest-quality masks per object from the resulting tracks as a fixed prompt set, and re-initialise $\mathcal{S}$ with it. We pick an anchor frame $t^\star = \mathrm{argmax}_t \sum_i \tilde{u}_t^{\mathrm{on},(i)} \mathbf{1}[\tilde{u}_t^{\mathrm{on},(i)} > \tau_u]$ from the online per-object IoU scores, and run PROPSTEP forward and backward from $t^\star$ with update_prompt = False. The curated prompt set and bi-directional propagation together recover masks on frames where the online pass missed an object or produced a low-quality mask.

B Additional implementation details

GMOS-S training setup. GMOS-S follows the same optimiser, schedule, and hardware as the GMOS proposer (Sec. 4.3). For the loss in Eq. A1, we set $\lambda^{\mathrm{S}}_{\mathrm{mask}}$=20, $\lambda^{\mathrm{S}}_{\mathrm{iou}}$=1, and $\lambda^{\mathrm{S}}_{\mathrm{conf}}$=0.5, with a rescaled confidence upper bound $c_{\max} = 5$. The frame-level mask loss combines focal loss (α=0.25, γ=2) and dice loss in a 20:1 ratio.

Propagator settings. For the thresholds used in Algorithm 1, we set $\tau_m = 0.5$ (motion score above which an object is considered moving), $\tau_u = 0.7$ (proposer IoU prediction above which a mask is trusted), $\tau_{\mathrm{match}} = 0.95$ (IoU between the proposer and propagated masks above which the proposer mask reinforces the existing track), and $\tau_{\mathrm{new}} = 0.3$ (maximum per-track precision below which the proposer mask is treated as a new object). For the offline SELECTTOPK step, we pick the top 10% of frames per object ranked by proposer IoU prediction, keeping only those with IoU above 0.95.

Runtime measurement. Per-frame runtimes in Tab. 1 are measured on a single NVIDIA RTX A6000 GPU over all frames of DAVIS17. Each measurement includes preparation of auxiliary input modalities (optical flow, depth, or point tracks), the model forward pass, and any post-processing (*e.g.* SAM2 refinement), but excludes data-loading overhead.

Algorithm 1 Pseudo-code of the GMOS propagator.

Thresholds (values in Appendix B):
τ_m: an object is moving if its motion score exceeds this.
τ_u: the proposer IoU prediction is trusted if it exceeds this.
τ_{match}: proposer and propagated masks refer to the same track if their IoU exceeds this.
τ_{new}: a proposer mask is treated as a new track if its max precision w.r.t. any propagated mask is below this.

Helper functions:
$\textsc{Precision}(A,B) = |A \cap B|/|A|$: the fraction of mask A covered by mask B.
$\textsc{HungarianMatch}(\tilde{M}_t, \{\hat{M}_t, \hat{m}_t, \hat{u}_t\})$: solve a bipartite assignment between propagated masks $\tilde{M}_t$ and proposer masks $\hat{M}_t$ on mask IoU, returning the proposer predictions reordered to align with the propagated tracks.

SAM2 operations:
$\textsc{InitSAM2}(\{(t,i,M_t^{(i)})\})$: initialise a SAM2 state $\mathcal{S}$ with a set of mask prompts, each tagging object i at frame t with mask $M_t^{(i)}$.
$\textsc{SAM2Propagate}(\mathcal{S},t)$: propagate the tracks currently held in $\mathcal{S}$ to frame t, returning the propagated masks $\tilde{M}_t$.
$\textsc{AddPrompt}(\mathcal{S},t,i,M)$: inject mask M as an additional prompt for object i at frame t into $\mathcal{S}$.

Elementary step
step $\textsc{PropStep}(\mathcal{S}, t_{\text{start}}, \text{direction}, \text{update_prompt})$ ▷ SAM2 state $\mathcal{S}$, starting frame, propagation direction $\in \{+1,-1\}$, whether to update the prompt set during propagation
  for $t \leftarrow t_{\text{start}}$, stepping by direction
    $\tilde{M}_t \leftarrow \textsc{SAM2Propagate}(\mathcal{S},t)$ ▷ propagated masks at frame t
    $\{\hat{M}_t, \hat{m}_t, \hat{u}_t\} \leftarrow \textsc{Proposer}(I_{t-n},\ldots,I_{t+n})$ ▷ mask, motion, IoU predictions
    $\{\hat{M}_t, \hat{m}_t, \hat{u}_t\} \leftarrow \textsc{HungarianMatch}(\tilde{M}_t, \{\hat{M}_t, \hat{m}_t, \hat{u}_t\})$
    if update_prompt
      for each object i with $\hat{m}_t^{(i)} > \tau_m$ **and** $\hat{u}_t^{(i)} > \tau_u$
        if $\max_j \textsc{Precision}(\hat{M}_t^{(i)}, \tilde{M}_t^{(j)}) < \tau_{\text{new}}$
          $\mathcal{S} \leftarrow \textsc{AddPrompt}(\mathcal{S}, t, i_{\text{new}}, \hat{M}_t^{(i)})$ ▷ *add* new track
        else if $\textsc{IoU}(\hat{M}_t^{(i)}, \tilde{M}_t^{(i)}) > \tau_{\text{match}}$
          $\mathcal{S} \leftarrow \textsc{AddPrompt}(\mathcal{S}, t, i, \hat{M}_t^{(i)})$ ▷ *reinforce* existing track i
    $\tilde{m}_t^{(i)} \leftarrow \mathbf{1}\left[\hat{m}_t^{(i)} > \tau_m\right]$ ▷ motion label; tracks without a moving match are labelled static
    store $(\tilde{M}_t^{(i)}, \tilde{m}_t^{(i)}, \tilde{u}_t^{(i)})$ for all i
  return $\mathcal{S}$, stored tracks

Online procedure: a single forward pass with injection
procedure ONLINE
  $\mathcal{S} \leftarrow \textsc{InitSAM2}(\{(1,i,\hat{M}_1^{(i)}) : \hat{m}_1^{(i)} > \tau_m, \hat{u}_1^{(i)} > \tau_u\})$
  $\mathcal{S}, \mathcal{T}^{\text{on}} \leftarrow \textsc{PropStep}(\mathcal{S}, 1, +1, \text{update_prompt} = \text{True})$
  return $\mathcal{T}^{\text{on}}$

Offline procedure: refines online output with a bi-directional second pass
procedure OFFLINE
  $\mathcal{T}^{\text{on}} \leftarrow \textsc{Online}$
  $P \leftarrow \textsc{SelectTopK}(\mathcal{T}^{\text{on}})$ ▷ per-object top-K high-quality masks
  $\mathcal{S} \leftarrow \textsc{InitSAM2}(P)$
  $t^\star \leftarrow \text{argmax}_t \sum_i \tilde{u}_t^{\text{on},(i)} \mathbf{1}[\tilde{u}_t^{\text{on},(i)} > \tau_u]$ ▷ anchor frame from online IoU scores
  $\mathcal{S}, \mathcal{T}_+^{\text{off}} \leftarrow \textsc{PropStep}(\mathcal{S}, t^\star, +1, \text{update_prompt} = \text{False})$
  $\mathcal{S}, \mathcal{T}_-^{\text{off}} \leftarrow \textsc{PropStep}(\mathcal{S}, t^\star, -1, \text{update_prompt} = \text{False})$
  return $\mathcal{T}_+^{\text{off}} \cup \mathcal{T}_-^{\text{off}}$

Table C1: **Overall dataset statistics.** Asterisks (*) denote subsets in GMOS-2K.

Split	Dataset	Videos	Duration (hr)	Anno. frames	Anno. frames / video	Objects	Objects / video	Motion prop. /object
Train (Synthetic)	Kubric [22]	10,000	5.56	240,000	24.0	29,294	2.93	60.8%
	PointOdyssey [89]	43	1.24	107,514	2500.0	64	1.49	57.1%
	DynamicReplica [27]	483	1.68	144,900	300.0	933	1.93	60.7%
Train (Real)	*DAVIS17 [58]	44	0.03	3,101	70.5	92	2.09	94.8%
	*MoCA-Mask [11]	26	0.11	1,978	76.1	26	1.00	54.8%
	*HOI4D [45]	273	0.95	81,900	300.0	609	2.23	69.9%
	*OVIS [59]	404	0.24	25,925	64.2	1,865	4.62	92.6%
	*YTVOS19 [78]	1,183	1.74	31,327	26.5	1,623	1.37	91.1%
	Mannequin Challenge [39]	1,715	1.22	145,483	84.8	0	0.00	0.0%
Total (Train)		**14,171**	**12.77**	**782,128**	**55.2**	**34,506**	**2.43**	**64.2%**
Test	*DAVIS17 [58]	19	0.01	1,237	65.1	31	1.63	96.9%
	*YTVOS19 [78]	261	0.40	7,115	27.3	402	1.54	91.5%
Total (Test)		**280**	**0.41**	**8,352**	**29.8**	**433**	**1.55**	**91.9%**

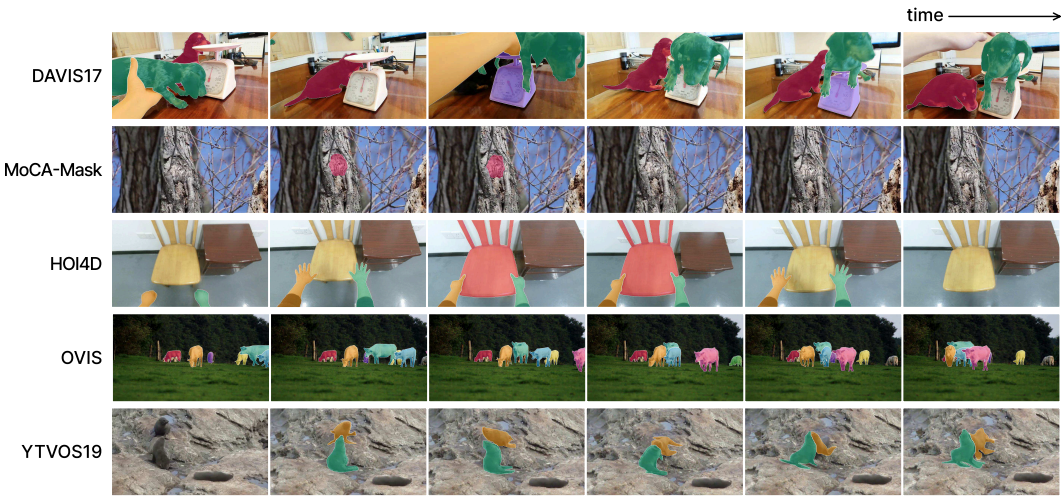

Figure C2: **GMOS-2K example sequences.** Videos are sampled from each of the five constituent subsets, with ground-truth segmentation masks overlaid. Frames are sampled uniformly along the temporal axis (left to right). An object is masked only in frames where it is actively moving, reflecting the temporally fine-grained motion labels in GMOS-2K.

C Dataset details

Overall dataset statistics. As described in Sec. 3, our training set comprises three synthetic datasets (Kubric, PointOdyssey, and DynamicReplica), five real-video subsets from GMOS-2K, and the Mannequin Challenge dataset [39], which contains no moving objects and serves as a source of static-scene negatives. For temporally fine-grained MOS evaluation, we use the DAVIS17 and YTVOS19 subsets of GMOS-2K as test sets. All datasets are open-sourced to research usage, with licence and terms in the datasets strictly respected.

Tab. C1 reports the full statistics of each dataset, including the number of videos, total duration, annotated frames, object count, and the proportion of frames containing motion. Subsets belonging to GMOS-2K are marked with an asterisk (*). In total, the training set spans 14,171 videos with 782,128 annotated frames and 34,506 objects.

Example visualisations in GMOS-2K. Fig. C2 presents example sequences from each of the five GMOS-2K subsets, illustrating the diversity of scenes and motion patterns. An object is masked only in frames where it is actively moving, reflecting the temporally fine-grained nature of annotated motion labels. For instance, in HOI4D, two hands are in motion throughout most of the sequence, whereas the chair is masked only during the brief interval when it is being picked up. Similarly, the YTVOS19 example shows that the seal in the bottom-right corner is never masked, as it remains stationary throughout the sequence.

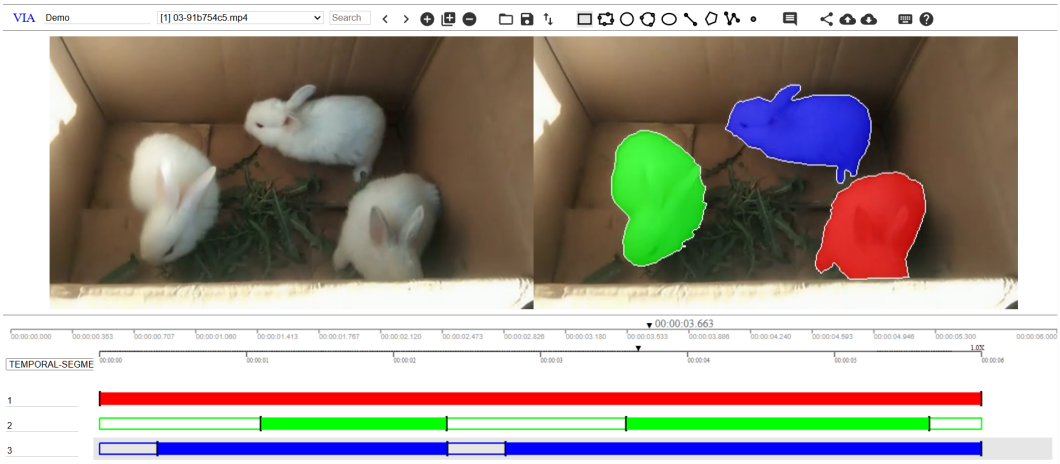

Figure D3: **VIA interface for Temporally Fine-grained Annotation (TFA).**

D GMOS-2K curation details

D.1 Instructions and tools for annotators

Video-level filtering. As described in Sec. 3, the curation pipeline begins by filtering 5,001 candidate videos through two sequential questions: (*i*) whether all independently moving objects have ground-truth segmentation masks, and (*ii*) whether all annotated objects are moving throughout the entire sequence. Annotators review each video and label their answers accordingly. The detailed instructions provided to annotators are shown below.

Video-level filtering instructions

For each video, please answer the following two questions:

Question 1: Do all moving objects in the video have corresponding segmentation masks?

- You may ignore all stationary objects (only moving objects need to be checked).
- **Yes** → label as `1`
- **No** → label as `-1`

Question 2: Are all segmented objects moving in every single frame of the video?

- **Yes** (every segmented object is moving in every frame) → label as `1`
- **No** (at least one segmented object is stationary in some frame, or all are stationary) → label as `-1`

Note: If Question 1 is labeled `-1`, you may skip Question 2.

Temporally Fine-grained Annotation (TFA). Videos that pass the first filter but fail the second are forwarded for TFA, which labels per-object motion intervals along the temporal axis. We perform TFA using VIA [18], an open-source annotation tool with built-in temporal grounding functionality. As shown in Fig. D3, we adapt the default VIA interface with two modifications: (*i*) the original video and its mask-overlaid counterpart are displayed side-by-side, with each object assigned a unique colour for easy identification; and (*ii*) the temporal axes are colour-coded to correspond to individual objects. The detailed annotation workflow and instructions are provided below.

Temporally Fine-grained Annotation (TFA) instructions

(Fig. D3 provides an example VIA interface)

Step 1. Enter the VIA annotation tool and import a video to annotate.

Step 2. *(Optional)* Open "Keyboard Shortcuts" to learn VIA operations.

Step 3. Each object is associated with an automatically generated time axis identified by colour codes. For each object, label the temporal windows when the object *truly moves* (regardless of whether the camera is moving).

- Only focus on objects with segmentation masks; ignore motions of shadows, smoke, fluid, etc.
- When the object is invisible, treat it as "static" — do not label.
- If the object moves all the time, label the full time axis. If the object is always static or invisible, leave its time axis empty.

Step 4. Click the "Export" button and save the CSV file.

Step 5. Return to Step 1.

Note: Step 4 (saving) can be done for a single video or multiple videos together.

Quality control. We apply quality control protocols at each of the above stages.

- *Video-level filtering.* Two annotators independently answer the two filtering questions for each candidate video. Sequences on which they disagree are flagged and jointly reviewed until a consensus is reached, ensuring that no qualifying video is dropped and no unsuitable video is forwarded to TFA.
- *Temporally Fine-grained Annotation (TFA).* After each sequence is annotated, a second annotator visualises the resulting MOS-I ground truth, with each object's mask overlaid only on the frames within its labelled motion intervals. The reviewer checks for (i) moving objects whose motion intervals are missing entirely (an object never receives a mask despite moving), (ii) static or background objects falsely labelled as moving (a mask appears on frames where the object is at rest), and (iii) inaccurate motion–rest transition boundaries. This step also captures errors from the filtering stage, removing any unsuitable videos that erroneously passed the previous filter. Flagged cases are returned to the original annotator for revision and re-verified by the reviewer.

  For transition boundaries, the reviewer independently labels the motion–rest boundaries and compares them with the original annotation. A clip is returned for revision if the two differ by more than two frames (approximately 0.083 s for videos recorded at 24 fps), a tolerance we consider reasonable given that the exact moment an object starts or stops moving is inherently somewhat subjective. This boundary refinement is iterative: after the first iteration, approximately 37% of the sequences exhibited boundary differences of more than two frames, and all such discrepancies were resolved in subsequent iterations.

D.2 Curation statistics

Per-subset statistics. Tab. D2 provides a detailed breakdown of the curation pipeline described in Sec. 3, reporting the number of videos retained after each filtering and annotation stage for every GMOS-2K subset. Of the $5{,}001$ initial candidate videos, $2{,}791$ are discarded by the first video-level filter, and the remaining $2{,}210$ form the final dataset, of which $1{,}467$ require TFA.

Table D2: **Curation statistics for GMOS-2K.** "TFA" stands for Temporally Fine-grained Annotations.

Split	Dataset	Initial videos	Filtered videos	Videos in GMOS-2K		
				w/o TFA	w/ TFA	Total
Train	DAVIS17 [58]	60	16	32	12	44
	MoCA-Mask [11]	26	0	2	24	26
	HOI4D [45]	300	27	0	273	273
	OVIS [59]	607	203	59	345	404
	YTVOS19 [78]	3,471	2,288	526	657	1,183
Total (Train)		**4,464**	**2,534**	**619**	**1,311**	**1,930**
Test	DAVIS17 [58]	30	11	15	4	19
	YTVOS19 [78]	507	246	109	152	261
Total (Test)		**537**	**257**	**124**	**156**	**280**

Annotation cost. Three annotators are employed for the entire curation process. Stage 1 (video-level filtering) takes roughly 17 hours to check for all $5{,}001$ candidates. Stage 2 (TFA) requires approximately

Table E3: **Training dataset composition.** "M.C." is short for Mannequin Challenge.

Syn.	GMOS -2K	M.C.	DAVIS17-IM		
			mtIoU↑	$\mathcal{J}_{\text{mov}}$ ↑	FP count↓
✓	✗	✗	53.0	65.0	**0.097**
✓	✓	✗	67.4	82.1	0.170
✓	✓	✓	**68.7**	**82.2**	0.147

Table E4: **Confidence-based loss.**

Confidence loss	DAVIS17-IM		
	mtIoU↑	$\mathcal{J}_{\text{mov}}$ ↑	FP count↓
✗	65.4	81.9	0.192
✓	**68.7**	**82.2**	**0.147**

Table E5: **Input temporal window size for GMOS proposer.** *Our default setup takes an input temporal window of 0.5s.

Temporal window (s)	DAVIS17-IM		
	mtIoU↑	$\mathcal{J}_{\text{mov}}$ ↑	FP count↓
0.16	67.1	79.4	**0.130**
0.50*	**68.7**	82.2	0.147
1.00	66.7	**82.7**	0.197

Table E6: **Input frame gap for GMOS proposer.** *Our default setup takes the input frame gap of "1 1 1 1".

Frame gap	DAVIS17-IM		
	mtIoU↑	$\mathcal{J}_{\text{mov}}$ ↑	FP count↓
1 1 1 1*	68.7	82.2	**0.147**
2 1 1 2	68.1	82.9	0.155
3 1 1 3	**68.9**	**83.4**	0.180

Table E7: **Fusing early-layer π^3 features in GMOS proposer.** *Our default setup does not take additional early-layer π^3 features.

Early-layer π^3 feat.	DAVIS17-IM		
	mtIoU↑	$\mathcal{J}_{\text{mov}}$ ↑	FP count↓
feat.	**69.0**	**82.3**	0.142
attn.	68.8	82.1	**0.126**
✗*	68.7	82.2	0.147

Table E8: **Mask decoder in GMOS proposer.** *Our default setup adopts a trainable mask decoder.

Mask decoder	DAVIS17-IM		
	mtIoU↑	$\mathcal{J}_{\text{mov}}$ ↑	FP count↓
SAM2	60.4	78.2	0.211
Trainable*	**68.7**	**82.2**	**0.147**

170 hours of annotation for the 1,467 videos that need temporal labels. An additional 20 hours are spent on quality verification and re-annotation, bringing the total annotation effort to approximately 207 person-hours. All annotators are compensated above the local minimum wage.

E Further ablation study

E.1 Ablations on GMOS proposer

To probe the proposer directly, we skip the propagator and apply per-frame Hungarian matching between proposer outputs and the ground truth, reporting performance on DAVIS17-IM.

Training dataset composition. Tab. E3 investigates how the major subsets contribute to training. Synthetic data alone yields a strong proposer with a low FP count, but limited mtIoU and $\mathcal{J}_{\text{mov}}$ due to the synthetic-to-real domain gap. Adding the real moving-object data from GMOS-2K substantially improves segmentation quality, while the static-scene Mannequin Challenge clips further reduce the FP count by teaching the model to suppress false motion induced by depth parallax.

Confidence-based loss. Tab. E4 shows that the confidence-based loss improves all three metrics by mitigating noisy motion annotations near static–moving transitions.

Inference temporal window size. Tab. E5 varies the temporal window from which five input frames are uniformly sampled at inference. Since the proposer is trained with variable window lengths, performance remains relatively stable across settings. In particular, a larger window makes moving objects easier to discover (raising $\mathcal{J}_{\text{mov}}$) but reduces temporal granularity, resulting in more false positives. A window of ~ 0.5 s strikes the best balance, achieving the highest mtIoU.

Table E9: **Propagation strategies in GMOS propagator.** "Prop. 1" and "Prop. 2" denote the start-frame strategy for the first and second propagation stages. *forward* starts at frame 1, while *select* starts at the anchor frame with the highest aggregate predicted IoU on confident moving proposals. "MOS-I" and "MOS" indicate which evaluation protocols each configuration supports.

Exp.	Prop. 1	Prop. 2	Online	MOS-I	MOS	DAVIS17-IM		
						mtIoU↑	$\mathcal{J}_{\text{mov}}$↑	FP count↓
A *(online default)*	forward	-	✓	✓	✗	72.9	83.5	0.075
B	forward	forward	✗	✓	✓	72.4	83.7	0.080
C	select	forward	✗	✓	✓	73.3	82.9	0.039
D	select	select	✗	✓	✓	74.1	84.0	0.043
E *(offline default)*	forward	select	✗	✓	✓	72.9	84.2	0.079

Input frame gap. Tab. E6 replaces the uniformly sampling frame gap (*i.e.* "1 1 1 1") with non-uniform gaps that emphasise the centre frame. Non-uniform gaps slightly improve $\mathcal{J}_{\text{mov}}$ on moving objects but consistently raise FP count, suggesting that uniform sampling provides the most balanced motion cues for the proposer.

Fusing early-layer π^3 features. By default, the GMOS proposer takes two feature sources as input, geometric (the last layer of the π^3 encoder) and segmentation (the last layer of the SAM2 encoder). We investigate whether adding a *third* source, early-layer signals from the π^3 encoder, improves the proposer. As outlined in Tab. E7, we consider two modes, both sampling early-layer π^3 signals from layers 7, 15, 23, and 31 (of 36). In *feature* mode, activations from the sampled layers are concatenated and projected. In *attention* mode, cross-attention maps between the middle frame and all other frames are concatenated and projected to form the additional signal.

Tab. E7 shows that both variants perform on par with the default, indicating that the final-layer π^3 feature alone already suffices for motion reasoning. We therefore retain only the final-layer π^3 feature as the geometric input.

Mask decoder. As detailed in Sec. 2.1, the GMOS proposer employs a trainable hypernetwork mask decoder, in which the feature map is upsampled and attended by object embeddings.

An alternative is to discard the upsampler and predict masks directly at the bottleneck resolution (64×64), using off-the-shelf SAM2 to recover the full-resolution mask for each object. Specifically, we randomly sample 10 positive and 3 negative points from the low-resolution prediction of each object as prompts to SAM2.

Tab. E8 compares our lightweight trainable upsampling decoder against this SAM2-based decoding. The latter underperforms on all metrics, suggesting that the low-resolution prediction does not fully capture moving-object information.

E.2 Ablations on GMOS propagator

Tab. E9 investigates the start-frame strategy for each propagation stage in the GMOS propagator. Exp. A corresponds to our default online setup, with a single forward propagation starting at frame 1. Since the sequence is traversed only once in temporal order, masks cannot be recovered for every frame (*e.g.* an object initially static but moves later). Therefore, GMOS (online) is restricted to the MOS-I setup.

The four remaining variants (Exp. B–E) add a second propagation stage, enabling both MOS-I and MOS evaluation. Among these, Exp. C and D start the first propagation from a selected anchor frame and perform slightly better overall, but they require access to the full sequence to select the anchor frame before the first propagation stage. We therefore adopt Exp. E as our offline default, since its first stage coincides with the online procedure and its second stage further refines the result via a curated prompt set.

E.3 Repeating experiments

The training of GMOS proposer is stochastic, while the propagator step is fully deterministic. We repeat the proposer training with four different random seeds (Exp. A–D), report the per-frame Hungarian-matched results in the left half of Tab. E10, and apply the deterministic propagator on top of each to obtain the sequence-level (object ID associated) results in the right half. The small standard

Table E10: **Repeated runs of GMOS on DAVIS17-IM.** Exp. A–D train the proposer with four random seeds, and the deterministic propagator is then applied on top of each.

Exp.	DAVIS17-IM (GMOS proposer)			DAVIS17-IM (GMOS propagator)		
	mtIoU↑	$\mathcal{J}_{\text{mov}}$↑	FP count↓	mtIoU↑	$\mathcal{J}_{\text{mov}}$↑	FP count↓
A	68.7	82.2	0.147	72.9	84.2	0.079
B	69.2	81.9	0.147	72.4	84.1	0.077
C	69.0	82.6	0.135	73.7	85.5	0.045
D	68.3	83.4	0.133	71.7	85.1	0.081
Avg.	$68.8_{\pm 0.4}$	$82.5_{\pm 0.7}$	$0.141_{\pm 0.008}$	$72.7_{\pm 0.8}$	$84.7_{\pm 0.7}$	$0.071_{\pm 0.017}$

Table E11: **Streaming with a fixed delay vs. strictly causal inference.** *Default online setup (as in Tab. 4), whose centred window introduces a 0.25 s lookahead. The strictly causal variant predicts the mask for the last frame of the window and has no delay.

Method	YTVOS19-IM			DAVIS17-IM		
	mtIoU↑	$\mathcal{J}_{\text{mov}}$↑	FP count↓	mtIoU↑	$\mathcal{J}_{\text{mov}}$↑	FP count↓
GMOS (*online*, 0.25 s delay)*	64.2	**76.4**	0.197	**72.9**	**83.5**	**0.075**
GMOS (*strictly causal*)	**64.3**	74.2	**0.189**	69.9	81.4	0.135

Table E12: **MOS-I baselines equipped with a per-frame motion-state classifier.** The same classifier, trained on our temporally fine-grained annotations, is applied to all methods, so that no method benefits from additional motion-state supervision.

Method	YTVOS19-IM			DAVIS17-IM		
	mtIoU↑	$\mathcal{J}_{\text{mov}}$↑	FP count↓	mtIoU↑	$\mathcal{J}_{\text{mov}}$↑	FP count↓
FlowSAM [76]	21.1	62.8	3.085	22.7	75.3	4.101
FlowSAM [76] + motion classifier	44.6	62.5	0.454	60.7	74.9	0.299
SegAnyMo [24]	55.9	63.2	0.320	67.0	81.7	0.253
SegAnyMo [24] + motion classifier	58.4	62.6	0.208	69.3	81.4	0.128
GMOS	63.4	**76.6**	0.238	72.9	**84.2**	0.079
GMOS + motion classifier	**63.7**	75.0	**0.178**	**73.6**	83.6	**0.051**

deviations on both halves indicate that GMOS is statistically stable across seeds, and the propagator brings consistent gains over the proposer alone in every run.

E.4 Further analyses

Strictly causal inference. The online procedure (Sec. 2.2) traverses the sequence once in temporal order, but the proposer's centred five-frame window spans 0.5 s and thus includes two future frames (at $+0.125$ s and $+0.25$ s). The procedure is therefore streaming with a fixed delay of 0.25 s. We additionally evaluate a strictly causal variant, which uses a five-frame temporal window and predicts the mask for its *last* frame, thereby removing the delay. As shown in Tab. E11, the strictly causal variant performs on par on YTVOS19-IM, with a slightly higher mtIoU and lower FP count but a lower $\mathcal{J}_{\text{mov}}$, and drops by 3.0 mtIoU on DAVIS17-IM, while still outperforming all prior methods in Tab. 4. A purely causal window thus remains a viable option for latency-critical deployment.

Baselines with per-frame motion-state prediction. Existing MOS baselines treat motion as a sequence-level attribute (*e.g.* SegAnyMo [24] classifies each entire track as either moving or static) and therefore cannot predict per-frame motion states. To assess whether their MOS-I performance is limited merely by the lack of such supervision, we equip them with an independent per-frame motion-state classifier trained on the same data as GMOS (synthetic data and GMOS-2K). The classifier is based on GMOS-S and predicts a per-frame motion-probability mask. An object is labelled as moving in a given frame if the recall of its mask with respect to the motion mask exceeds 0.5. We apply the

Table E13: **Performance under large camera motion.** "All" denotes all sequences and "Large" the subset with large camera motion (camera path length > 1 m or rotation $> 45°$, estimated with Depth Anything 3 [42]).

Method	YTVOS19-M		DAVIS17-M		YTVOS19-IM		DAVIS17-IM	
	$\mathcal{J}\uparrow$		$\mathcal{J}\uparrow$		mtIoU↑		mtIoU↑	
	All	Large	All	Large	All	Large	All	Large
FlowSAM [76]	66.6	58.8	74.6	72.1	21.1	20.3	22.7	21.2
SegAnyMo [24]	64.6	58.1	77.4	74.0	55.9	52.8	67.0	64.7
GMOS w/o π^3 features	76.4	73.7	77.1	73.9	57.1	54.8	65.0	63.1
GMOS	**79.9**	**78.1**	**79.7**	**78.3**	**63.4**	**61.1**	**72.9**	**71.2**
Δ vs. prev. SoTA	+13.3	+19.3	+2.3	+4.3	+7.5	+8.3	+5.9	+6.5
Δ vs. w/o π^3 features	+3.5	+4.4	+2.6	+4.4	+6.3	+6.3	+7.9	+8.1

same classifier to FlowSAM, SegAnyMo, and GMOS, so that no method benefits from additional motion-state supervision.

As shown in Tab. E12, mtIoU and FP count improve substantially for both baselines, confirming that part of their original gap stems from the absence of per-frame motion prediction, while $\mathcal{J}_{\text{mov}}$ remains largely unchanged since the classifier mainly filters false-positive motion states. For GMOS, which is already trained with temporally fine-grained annotations, the classifier yields only marginal changes, and GMOS continues to outperform both classifier-equipped baselines on mtIoU and $\mathcal{J}_{\text{mov}}$.

Robustness to camera motion. To examine whether the 3D-aware geometric features are most beneficial under challenging viewpoint changes, we stratify the multi-object YTVOS19 and DAVIS17 benchmarks by the magnitude of camera motion. We use camera motion rather than depth parallax as the criterion, since parallax varies spatially within each sequence and is difficult to summarise by a single scalar, whereas camera motion can be measured reliably and provides a practical proxy for viewpoint variation. We estimate the camera extrinsics of each sequence with Depth Anything 3 [42] and label a sequence as having large camera motion if its total camera path length exceeds 1 m or its rotation exceeds $45°$, which selects approximately 34% of YTVOS19 and 58% of DAVIS17 sequences.

As shown in Tab. E13, the margin of GMOS over the previous state of the art widens on the large-camera-motion subset on all four benchmarks. Moreover, comparing GMOS with and without π^3 features, the gain from the geometric features is larger on the large-camera-motion subset on three of the four benchmarks and on par on YTVOS19-IM, confirming that the 3D-aware features contribute most under challenging camera motion.

F Additional qualitative visualisations

We provide additional qualitative results to complement Figs. 4 and 5 in the main paper. Fig. E4 shows further comparisons against representative baselines under the multi-object and foreground–background protocols, and Fig. E5 illustrates GMOS's behaviour on more in-the-wild sequences, distinguishing between MOS-I and MOS predictions.

G Discussion

Failure cases. Fig. E6 illustrates two representative failure modes, both stemming from the limited spatial granularity of the π^3 geometric backbone. The left example (a table football game) shows a scene with many fast-moving objects under heavy occlusion. GMOS misses several small moving instances in the background, suggesting that the backbone struggles with fine-grained moving-object discovery in cluttered scenes. The right example shows a driving scene in which a large reflective truck passes the camera. Although the truck is static, GMOS incorrectly predicts it as moving at a particular frame, exposing limitations of the backbone in per-object motion-state prediction under reflections and extreme depth parallax.

Possible extensions. We outline two natural directions for future work. First, the geometric backbone could be fine-tuned on larger and more diverse motion-annotated datasets to strengthen both fine-



Figure E4: **Additional qualitative comparison on MOS.** Example videos are sampled from YTVOS19 (first two columns), DAVIS17 (third column), and MoCA (last column). The middle block compares the multi-object methods, while the bottom block evaluates the foreground–background models.



Figure E5: **Additional GMOS results on in-the-wild videos.** Four in-the-wild sequences (sourced outside our training or test datasets) contrast GMOS's MOS and MOS-I predictions. Under MOS-I, only instantaneously moving objects are segmented at each frame, with object identities consistently associated across frames. Under MOS, GMOS produces full segmentation masks for every object that moves at any point in the sequence.

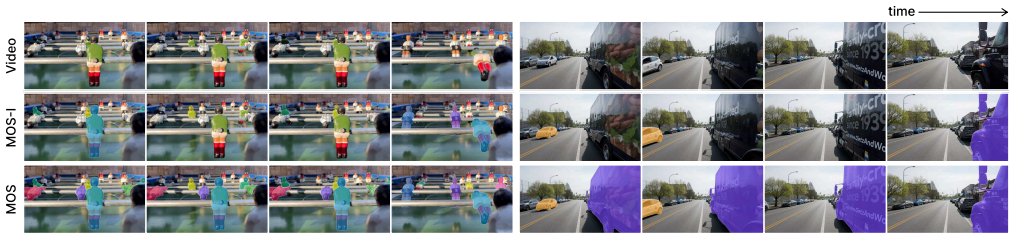

Figure E6: **Failure cases of GMOS. Left:** the model struggles when many moving objects are heavily occluded. **Right:** the model occasionally produces false positives in motion-state prediction.

grained moving-object discovery and motion-state prediction. Second, the proposer and propagator could be merged into a single end-to-end trainable framework, allowing the temporal consistency signal to improve mask quality and yield more accurate motion-state predictions over long videos.

Broader impact. GMOS enables real-time identification of moving objects directly from RGB video, supporting safety-critical applications such as autonomous driving, ecological and biomedical monitoring, public-safety surveillance, and dynamic 3D/4D scene reconstruction for robotics and AR/VR. Operating without pre-computed flow or trajectories also lowers the compute footprint of motion-aware perception. We release GMOS-2K and the MOS-I protocol for reproducible research. As with any video-segmentation system, GMOS could be misused in surveillance contexts that raise privacy concerns, particularly when combined with identity-recognition tools. We encourage downstream users to comply with data-protection regulations and obtain appropriate consent.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction enumerate four contributions, including the MOS-I evaluation protocol with three new metrics, the GMOS proposer-propagator framework, the GMOS-S variant, and the GMOS-2K dataset, which are each substantiated in Secs. 2 to 4. Quantitative claims (state-of-the-art on MOS, UVOS, and MOS-I, and over $3\times$ faster runtime) are supported by Tabs. 1 and 3 to 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The discussion in Appendix G elaborates limitations inherited from the frozen π^3 and SAM2 encoders. Representative failure cases are visualised and analysed in Fig. E6, covering both fine-grained moving-object discovery in cluttered scenes and motion-state mispredictions under reflections and parallax.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include theoretical results. Our contributions are empirical, comprising the GMOS method, the GMOS-2K dataset, and the MOS-I evaluation protocol with associated metrics.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sec. 4.3 specifies the architecture, optimiser, learning-rate schedule, training duration, hardware, loss weights, and input window setting. The propagator pseudo-code is given in Algorithm 1 with all thresholds listed in Appendix B. The GMOS-S variant is fully described in Appendix A including its loss weights. Dataset composition and per-subset statistics are given in Sec. 4.1 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The supplementary material includes a code repository with training and evaluation pipelines, together with temporally fine-grained GMOS-2K annotations indexed by dataset, split, and sequence name. Pretrained checkpoints will be released upon publication. In addition to the provided temporally fine-grained motion-state annotations, all videos and segmentation masks used for training and evaluation are publicly available and properly cited. Annotation tools and instructions are provided in Appendix D to ensure reproducibility of the dataset curation procedure.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and test details, including the optimiser, learning-rate schedule, loss weights, and training duration, are reported in Sec. 4.3. The evaluation benchmarks, data splits, and metrics are described in Sec. 4.2. Additional propagator thresholds and GMOS-S hyperparameters are provided in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: We report mean and standard deviation over four random training seeds for both the proposer and the propagator stages on DAVIS17-IM in Tab. E10 (Appendix E), confirming that GMOS is statistically stable across seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Sec. 4.3 reports training compute (2 NVIDIA RTX A6000 GPUs for approximately 2 days). Per-frame inference runtime is measured on a single NVIDIA RTX A40 GPU and reported in Tab. 1, with the measurement protocol detailed in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our research, including the curation of GMOS-2K from publicly available video segmentation datasets and the engagement of human annotators (whose compensation and instructions are reported in Appendix D), conforms to the Code in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: A discussion of positive and negative societal impacts is provided in Appendix G, covering positive applications (autonomous driving, camouflaged object detection, ecological and biomedical monitoring, public-safety surveillance, and 3D/4D scene reconstruction for robotics and AR/VR), the lowered compute footprint of operating directly on RGB video, and surveillance-related privacy concerns with recommended mitigations.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released model and dataset do not pose high risks for misuse. GMOS is a discriminative segmentation model and not a generative model, and GMOS-2K is curated entirely from existing publicly released video segmentation datasets that have already undergone their own content vetting.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All pretrained models (π^3, SAM2) and datasets (Kubric, PointOdyssey, DynamicReplica, DAVIS16, DAVIS17, YTVOS19, OVIS, MoCA, MoCA-Mask, HOI4D, SegTrackv2, FBMS-59, Mannequin Challenge) are properly cited on first use in Secs. 2 and 3 and Secs. 4.1 and 4.2. As stated in Appendix C, all datasets are open-sourced for research use, and the corresponding licences and terms of use are strictly respected.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We introduce two new assets: the GMOS-2K dataset and the GMOS model. GMOS-2K is documented through its curation pipeline (Sec. 3), per-dataset statistics (Appendix C, Tab. C1) and per-subset curation statistics (Appendix D, Tab. D2), example visualisations (Fig. C2), and annotation instructions (Appendix D). The GMOS model is documented in Sec. 2 and Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: Appendix D provides the full instructions given to annotators for both the video-level filtering and the temporally fine-grained annotation stages, together with a screenshot of the VIA annotation interface. The annotation cost paragraph reports the number of annotators, the total annotation effort, and confirms that all annotators are compensated above the local minimum wage.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: Our annotation work involves only routine video labelling on publicly released video datasets and does not pose risks to participants beyond those of standard data-labelling tasks. We follow the NeurIPS Code of Ethics and the guidelines of our institution, and ensure annotator compensation above the local minimum wage as detailed in Appendix D.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for assistance with writing, editing, and formatting, and did not influence the core methodology, scientific rigour, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.